Genetic Algorithms

Evolutionary computation

Is evolution intelligent?

Intelligence

- The capability of a system to adapt its behaviour to ever-changing environment.

Evolutionary computation simulates evolution on a computer.

- Results in a series of optimisation algorithms, usually based on a simple set of rules.
- Optimisation iteratively improves the quality of solutions until an optimal, or at least feasible, solution is found.

Evolutionary computation

Evolutionary computing

- Genetic algorithms
- Evolution strategies
- Genetic programming

All these techniques simulate evolution using

- Selection
- Mutation
- Reproduction

Genetic Algorithms

A genetic algorithm (GA) is a variant of stochastic beam search in which *successor states are generated by combining two parent states*, rather than by modifying a single state.

(Remember with beam searches we keep track of several best states, not a single one)

The analogy to natural selection is that the "successors" (offspring) of a pair of states (organisms) populate the next generation according to their "value" (fitness)

Genetic algorithms

Like beam search, GAs start with *k* randomly generated states (population)

Each state or individual is represented as a string over a finite alphabet (often a string of 0s and 1s)

We use evaluation function (fitness function) - higher values for better states.

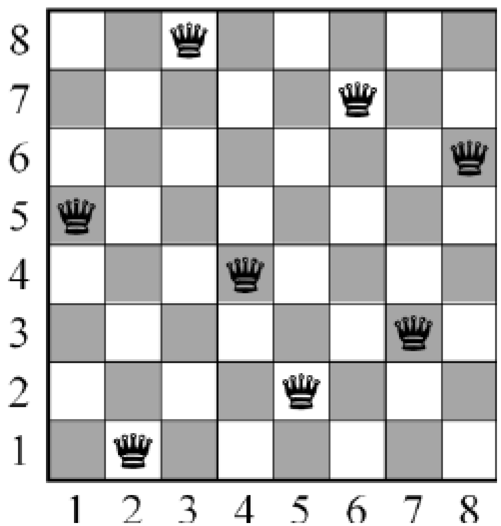
Produce the next generation of states by selection, crossover, and mutation. A successor state is generated by combining two parent states

GAs: Representing individuals

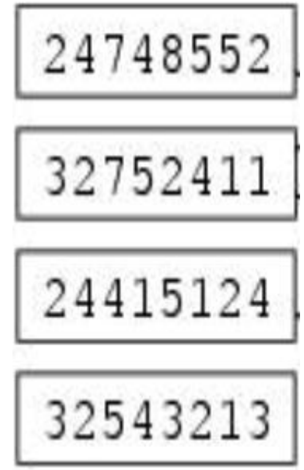
Artificial "chromosomes" consists of a number of "genes", and each gene is represented by 0 or 1:

1	0	1	1	0	1	0	0	0	0	0	1	0	1	0	1

GAs: Representing Individuals



5 1 8 4 2 7 3 6



(a)
Initial Population

For example, for the 8-queens problem the state could be represented as 8 digits string.

Each character in the string representing the position on one queen in the column, and it is in the range from 1 to 8.

Each string represents one possible state

Genetic algorithms

Two mechanisms link a GA to the problem it is solving:

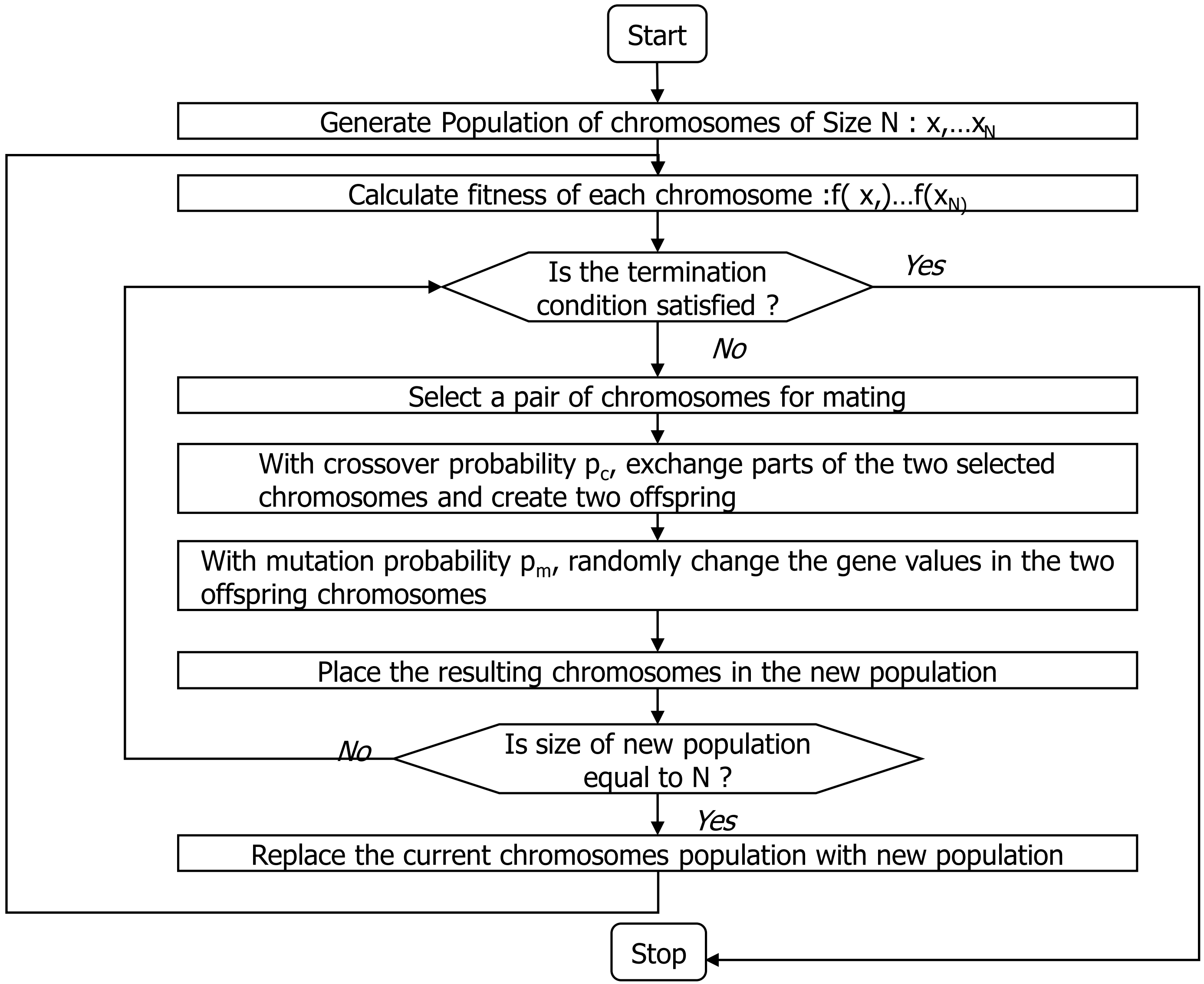
- Encoding
- Evaluation

GA use a measure of fitness of individual chromosomes to carry out reproduction.

As reproduction takes place

- Crossover operator exchanges parts of two single chromosomes
- Mutation operator changes the gene value in some randomly chosen location of the chromosome.
- Over time less fit chromosomes become extinct

Basic genetic algorithms



Basic genetic algorithms

Step 1:

- Represent the problem variable domain as a chromosome of a fixed length
- Choose the size of a chromosome population N, the crossover probability p_c and the mutation probability p_m.

Step 2:

- Define a fitness function to measure the performance, or fitness, of an individual chromosome in the problem domain.

Step 3:

- Randomly generate an initial population of chromosomes of size N: $x_1, x_2, \ldots, x_N$

Basic genetic algorithms

Step 4:

- Calculate the fitness of each individual chromosome:
  f (x1), f (x2), . . . , f (xN)

Step 5:

- Select a pair of chromosomes for mating from the current population.
- Parent chromosomes are selected with a probability related to their fitness.

Step 6:

- Create a pair of offspring chromosomes by applying the genetic operators – crossover and mutation.

Basic genetic algorithms

Step 7:

- Place the created offspring chromosomes in the new population.

Step 8:

- Repeat Step 5 until the size of the new chromosome population becomes equal to the size of the initial population, N.

Step 9:

- Replace the initial (parent) chromosome population with the new (offspring) population.

Step 10:

- Go to Step 4, and repeat the process until the termination criterion is satisfied.

Genetic algorithms

Each iteration is called a generation.

- Typical number of generations for a simple GA can range from 50 to over 500.

The entire set of generations is called a run.

At the end of a run expect to find one or more highly fit chromosomes

GAs use a stochastic search method therefore the fitness of a population may remain stable for a number of generations before a superior chromosome appears.

Common practice is to terminate the GA after a specified number of generations and then examine the population.

- If no satisfactory solution is found, the GA is restarted.

Genetic algorithms - example

Find the maximum value of the function ($15x – x^2$) where x varies between 0 and 15.

Assume that x takes only integer values. Thus, chromosomes can be built with only four genes:

Integer	*Binary code*	*Integer*	*Binary code*	*Integer*	*Binary code*
1	0 0 0 1	6	0 1 1 0	11	1 0 1 1
2	0 0 1 0	7	0 1 1 1	12	1 1 0 0
3	0 0 1 1	8	1 0 0 0	13	1 1 0 1
4	0 1 0 0	9	1 0 0 1	14	1 1 1 0
5	0 1 0 1	10	1 0 1 0	15	1 1 1 1

Size of the chromosome population N is 6

Crossover probability p_c equals 0.7

Mutation probability p_m equals 0.001.

The fitness function is defined by : $f(x) = 15x – x^2$

Genetic algorithms - example

Create initial population

- Six 4-bit strings with randomly generated ones and zeros

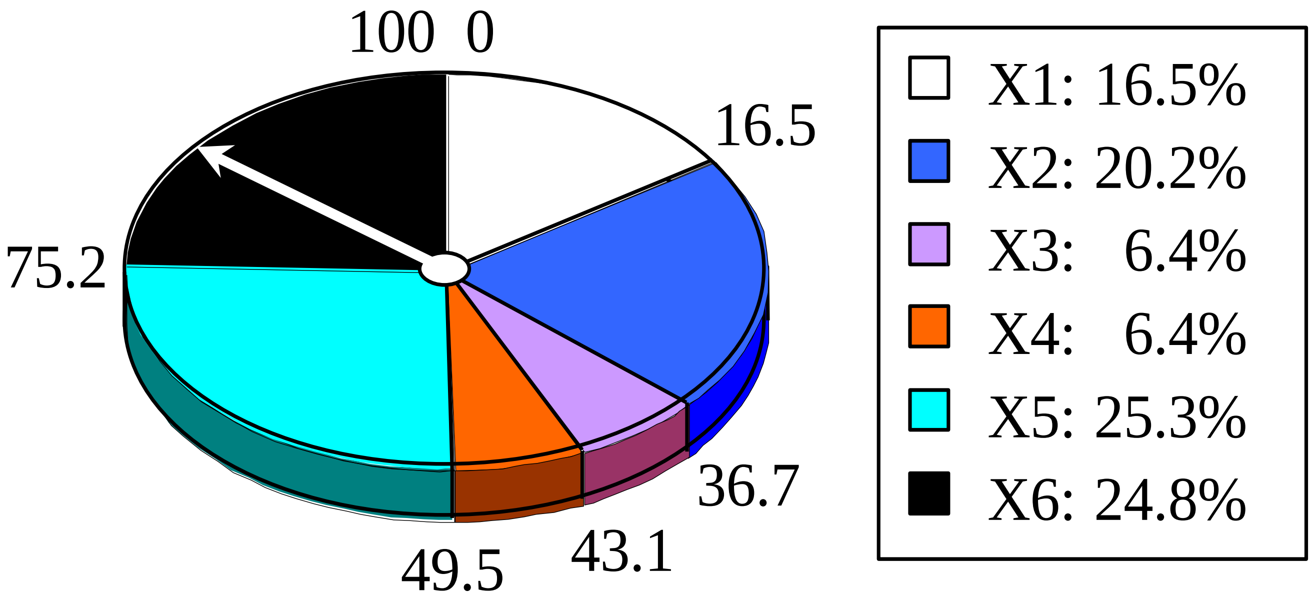
Calculate the fitness of each individual chromosome.

Chromosome label	*Chromosome string*	*Decoded integer*	*Chromosome fitness*	*Fitness ratio, %*
X1	1 1 0 0	12	36	16.5
X2	0 1 0 0	4	44	20.2
X3	0 0 0 1	1	14	6.4
X4	1 1 1 0	14	14	6.4
X5	0 1 1 1	7	56	25.7
X6	1 0 0 1	9	54	24.8

Roulette wheel selection

Chromosome selection techniques: roulette wheel selection

- The area corresponding to each chromosome is equal to the chromosome fitness ratio



Crossover operator

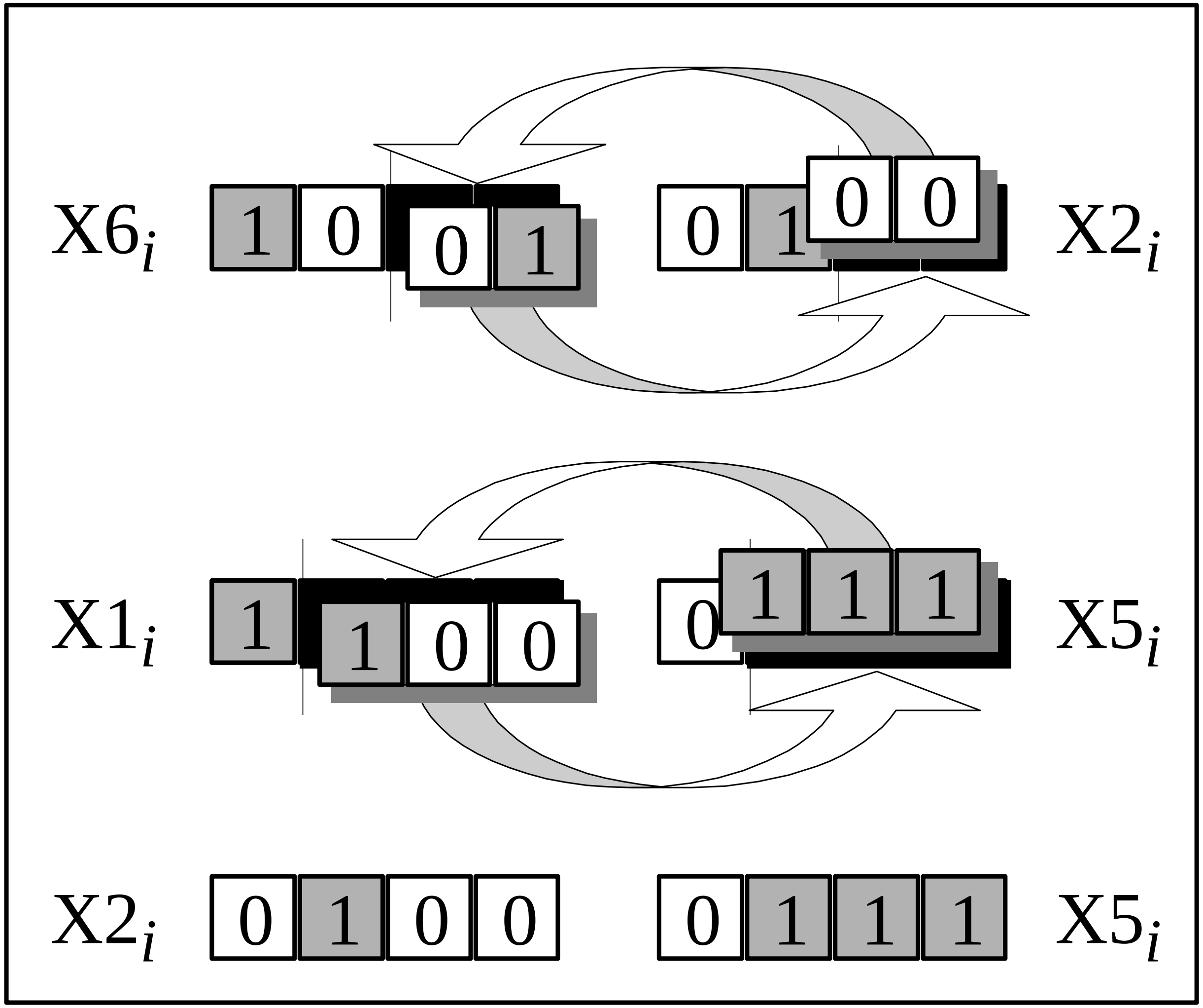
1. Randomly chooses a crossover point where two parent chromosomes "break"
2. Exchanges the chromosome parts after that point
3. Two new offspring are created.

If a pair of chromosomes does not cross over

- Chromosome cloning takes place
- Offspring are created as exact copies of each parent.

Typical value used for the crossover probability is 0.7.

Genetic algorithms – example crossover



Mutation operator

Mutation represents a change in the gene.

Mutation may lead to significant improvement or to worse results

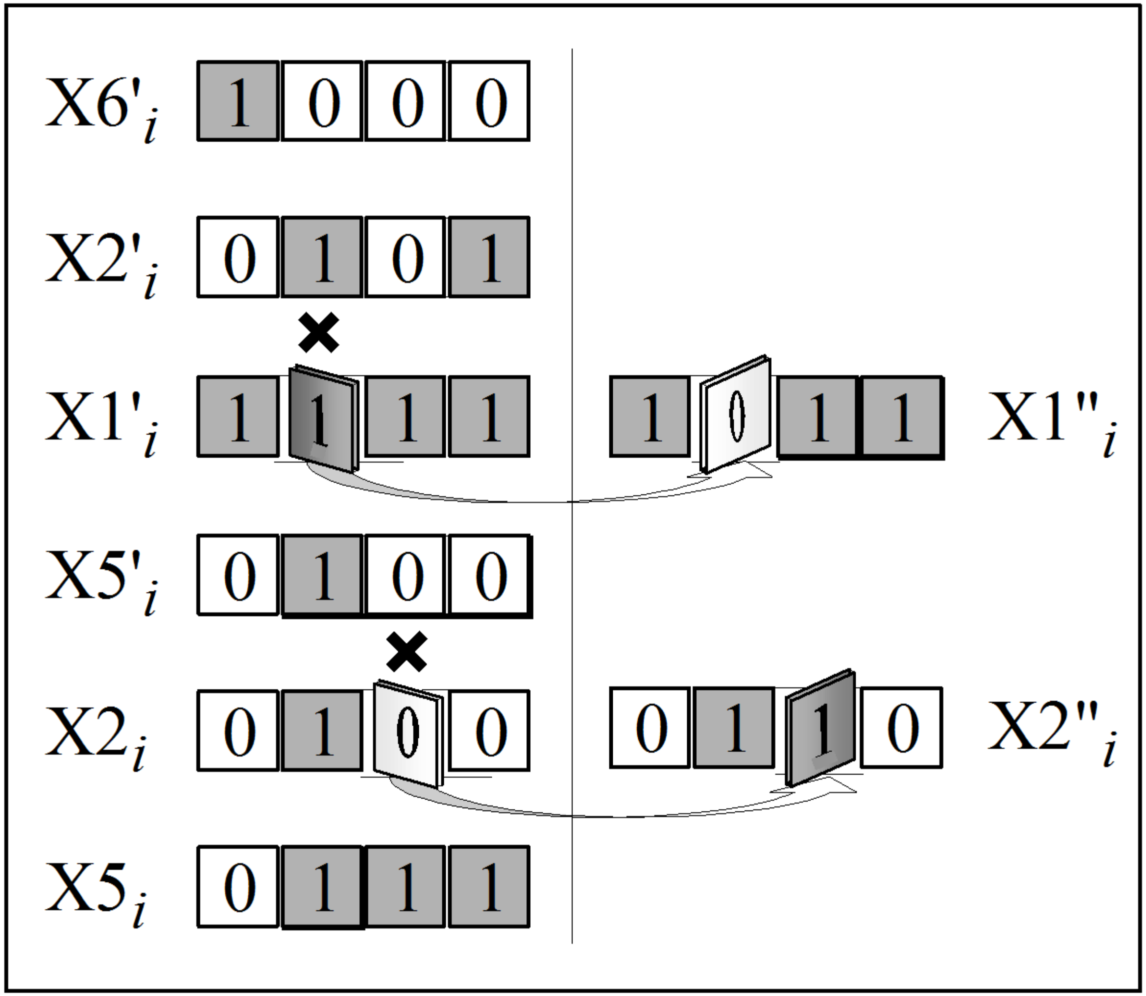
- Role is to provide a guarantee that the search algorithm is not trapped on a local optimum.

The mutation operator flips a randomly selected gene in a chromosome.

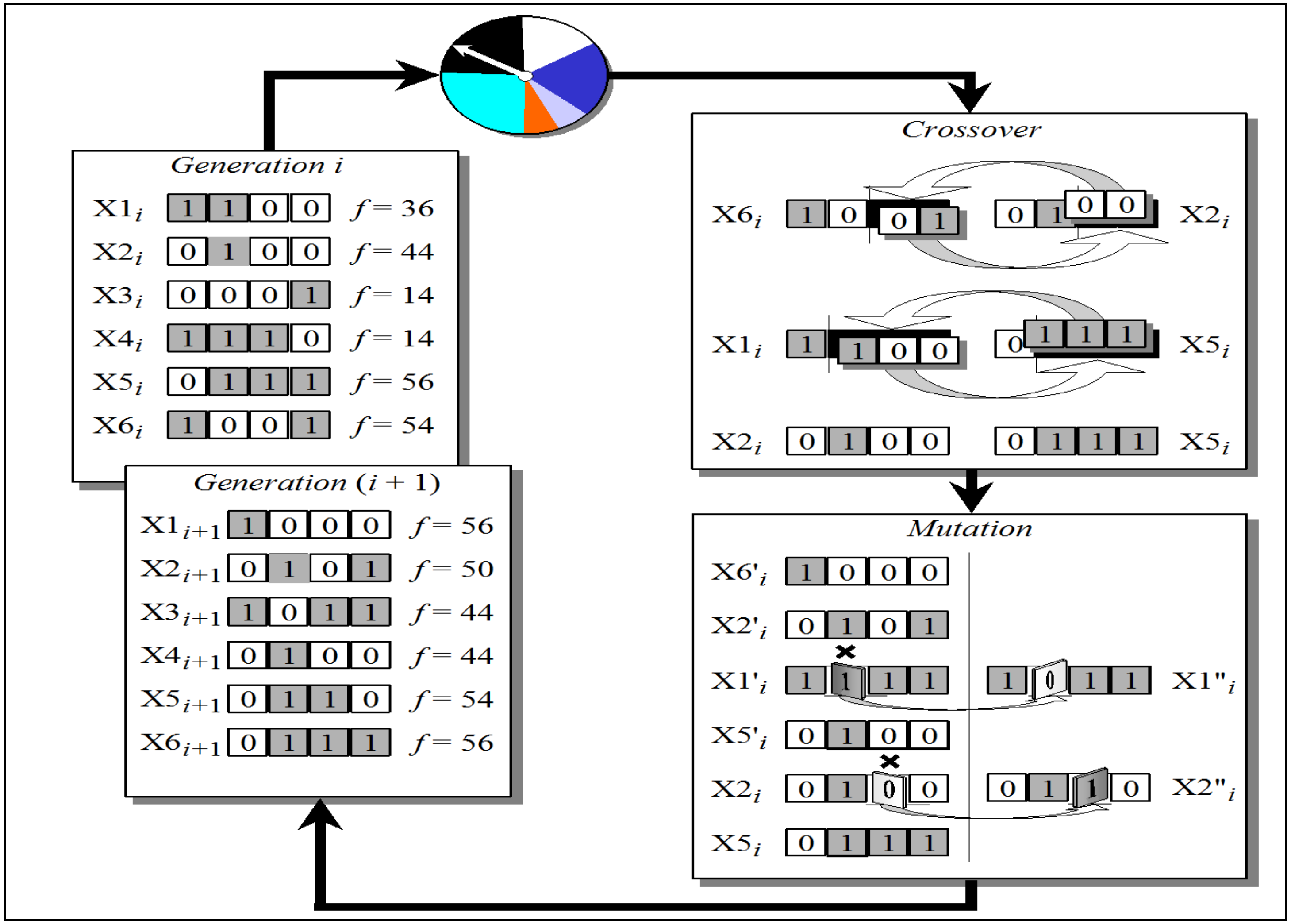
Mutation can occur in any gene of a chromosome with some possibility

The mutation probability is quite small in nature, and is kept low for GAs, typically in the range between 0.001 and 0.01.

Genetic algorithms - example mutation



Example - genetic algorithm cycle



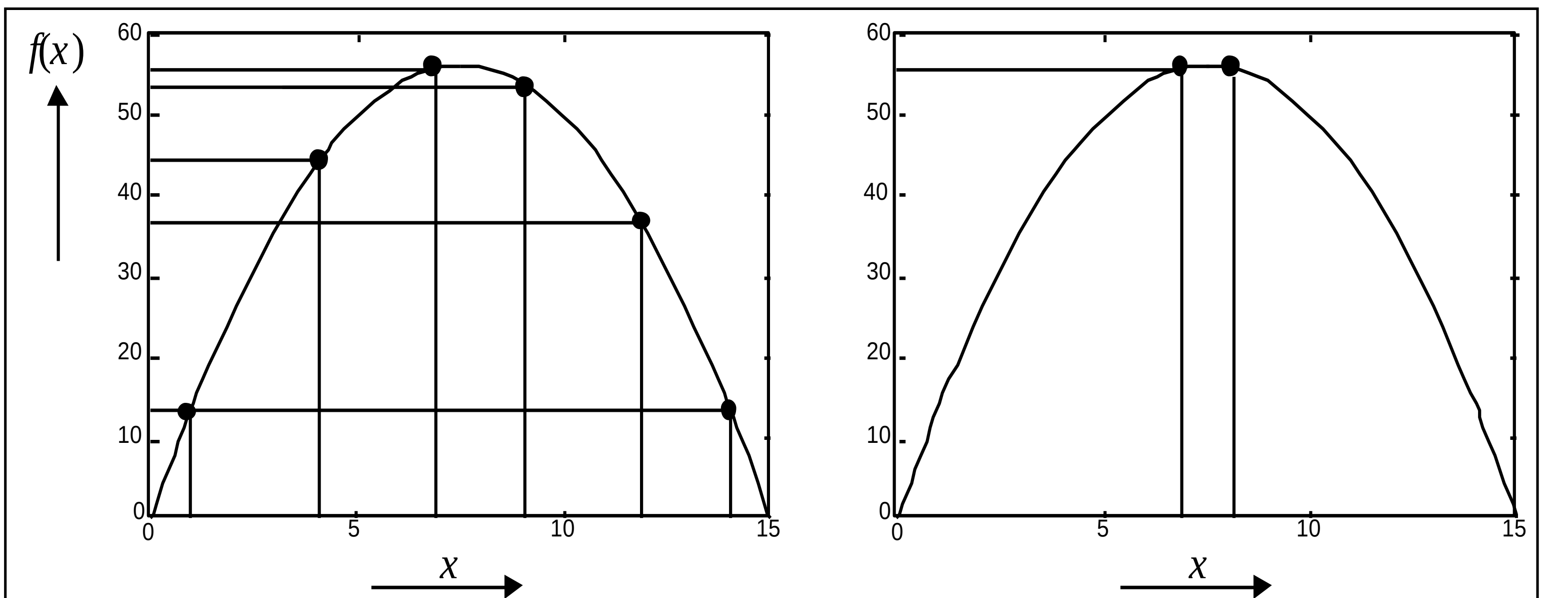
Genetic algorithms - result

After a number of generations (typically several hundred) the population evolves to a near-optimal solution.

In our example that would be the chromosomes

0	1	1	1

1	0	0	0



(a) Chromosome initial locations. (b) Chromosome final locations.

Genetic algorithms

Suppose we need to find the maximum of the "peak" function of two variables:

$$f(x, y) = (1 - x)^2 e^{-x^2 - (y+1)^2} - (x - x^3 - y^3)\, e^{-x^2 - y^2}$$

where parameters x and y vary between –3 and 3.

Represent the problem variables as a chromosome

- Represent parameters x and y as a concatenated binary string:

$$\underbrace{\boxed{1}\boxed{0}\boxed{0}\boxed{0}\boxed{1}\boxed{0}\boxed{1}\boxed{0}}_{x}\underbrace{\boxed{0}\boxed{0}\boxed{1}\boxed{1}\boxed{1}\boxed{0}\boxed{1}\boxed{1}}_{y}$$

Genetic algorithms

Choose the size of the chromosome population e.g. 6 and randomly generate an initial population.

Calculate the fitness of each chromosome. This is done in two stages.

- A chromosome, that is a string of 16 bits, is partitioned into two 8-bit strings:

1	0	0	0	1	0	1	0	and	0	0	1	1	1	0	1	1

- Then these strings are converted from binary (base 2) to decimal (base 10):

$$(10001010)_2 = 1\times 2^7 + 0\times 2^6 + 0\times 2^5 + 0\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0 = (138)_{10}$$

and

$$(00111011)_2 = 0\times 2^7 + 0\times 2^6 + 1\times 2^5 + 1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 = (59)_{10}$$

Genetic algorithms

Now the range of integers that can be handled by 8-bits, that is the range from 0 to ($2^8 - 1$), is mapped to the actual range of parameters x and y, that is the range from –3 to 3:

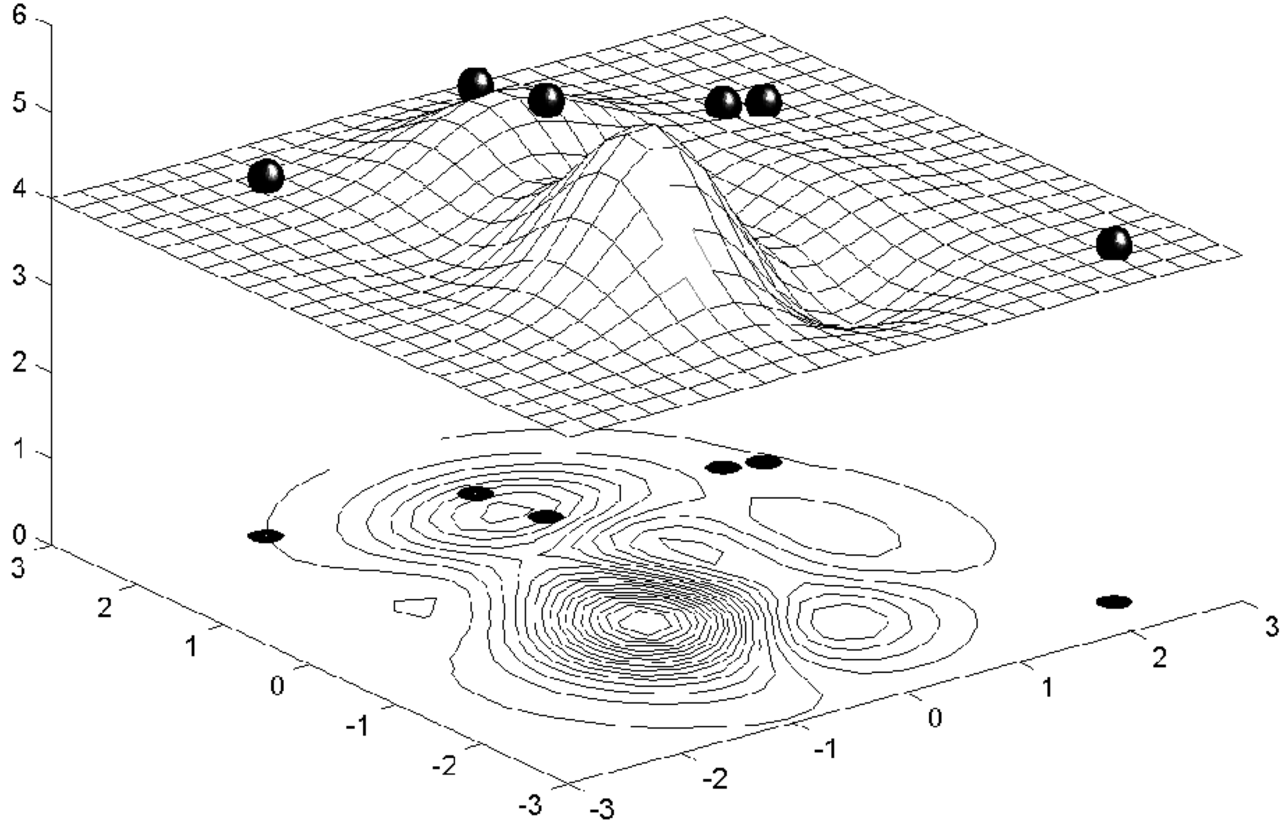
$$\frac{6}{256-1} = 0.0235294$$

To obtain the actual values of x and y, multiply their decimal values by 0.0235294 and subtract 3 from the results:

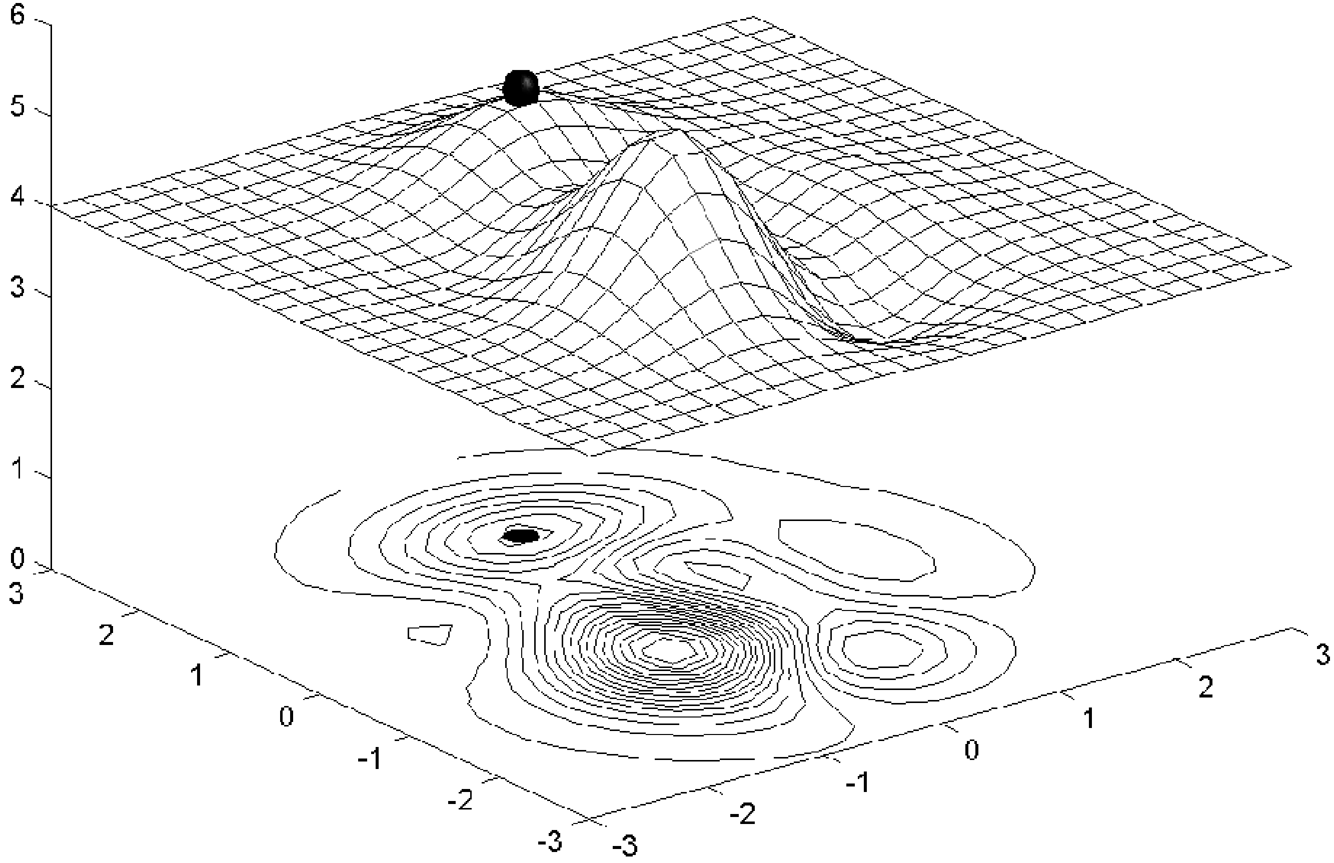
$x=(138)10 * 0.0235294 - 3 = 0.2470588$

$y =(59)10 * 0.0235294 - 3 = -1.6117647$

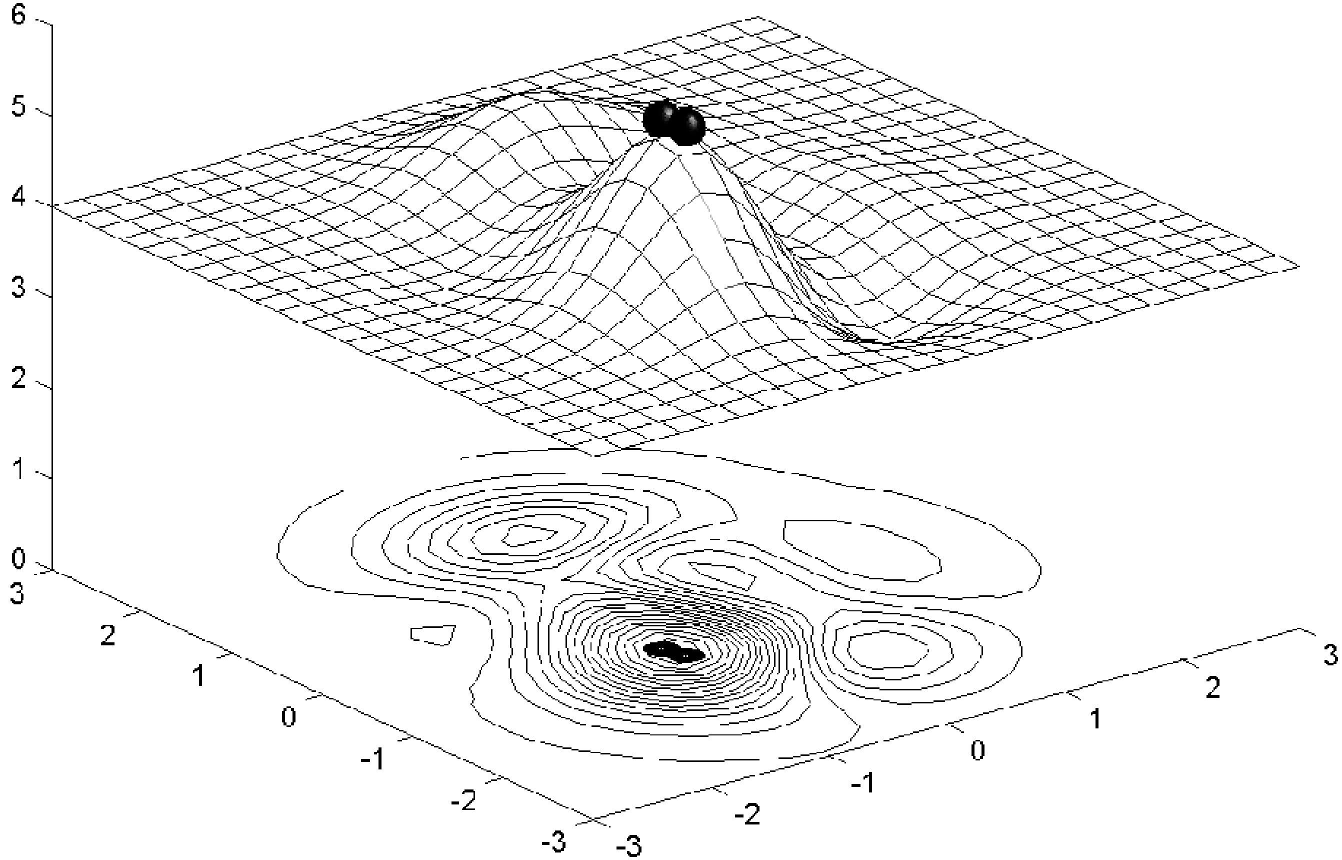
Chromosome locations on the surface of the "peak" function: initial population



Chromosome locations on the surface of the "peak" function: local maximum



Chromosome locations on the surface of the "peak" function: global maximum

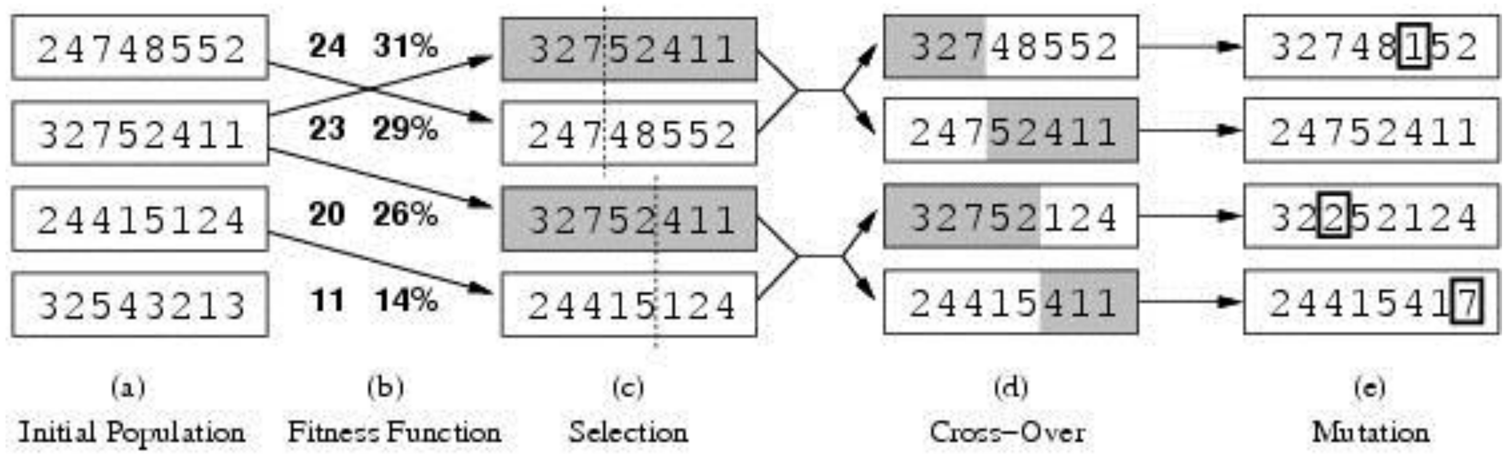


GAs: 8-Queens example

(a)
Initial Population

The representation we chose earlier for the 8-queens problem was strings of length 8, each element of the string representing the location of one queen in the relevant column.

GAs: Production of next generation

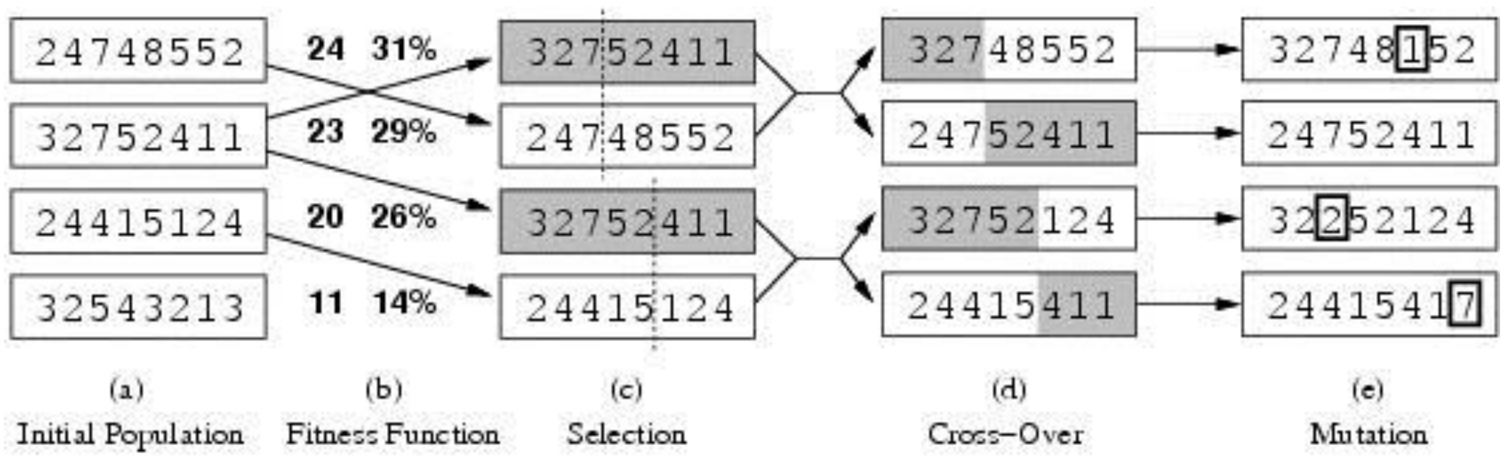


Figures (b) to (e) above illustrate the production of the next generation.

(b) Each state is rated by an evaluation function (fitness function in GA term)

- in this instance the fitness function is the number of *non-attacking* pairs of queens. This function has a value of 28 for a solution, and for the 4 states represented returns the values 24, 23, 20 and 11.
- In this variant of GA, the probability of being chosen for reproduction is directly proportional to the fitness score, and the percentages are shown next to the raw scores.

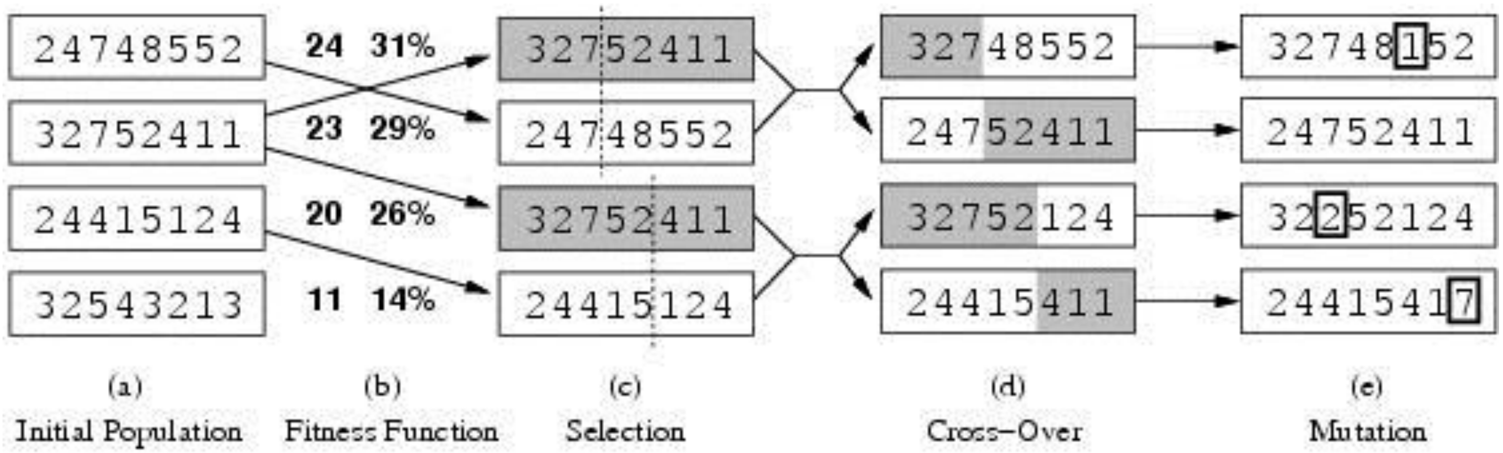
Genetic algorithms: Production of next generation



(c) A random choice of two pairs is selected for reproduction, in accordance with the probabilities in (b) (*one individual is selected twice and one not at all*).

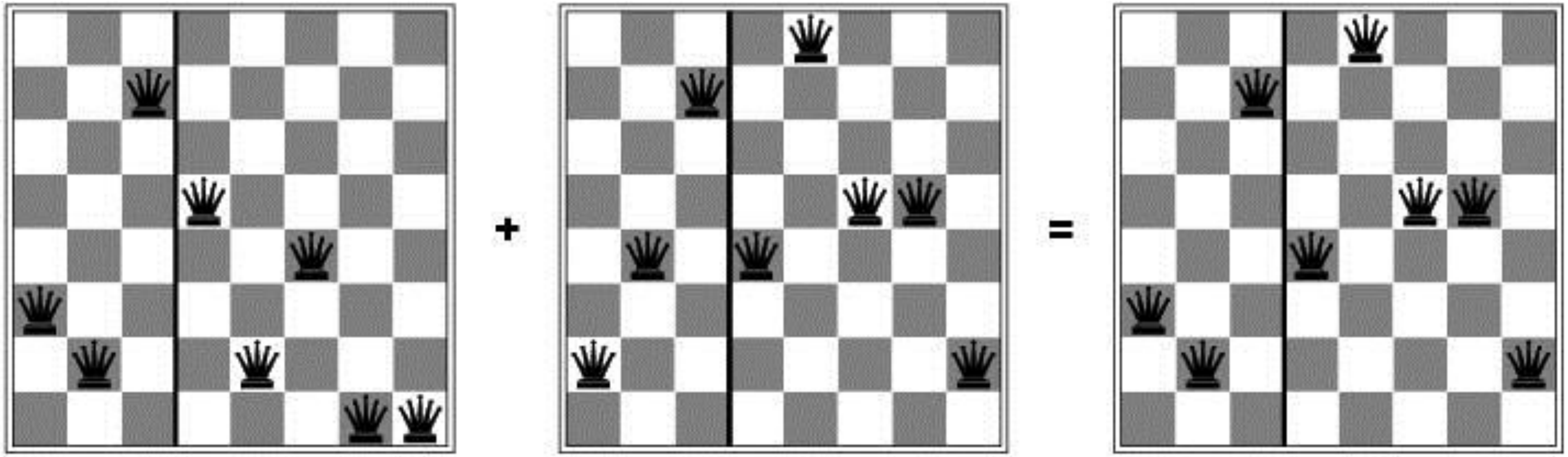
- For each pair to be mated a crossover point is randomly selected from the positions in the string. In the instances depicted above the points are after the third digit in the first pair and after the fifth digit in the second pair.

Genetic algorithms: Production of next generation



(d) The offspring themselves are created by crossing over the parent strings at the crossover point. For example, the first child of the first pair gets the first three digits from the first parent and the remaining digits from the second parent.

Genetic algorithms: Production of next generation

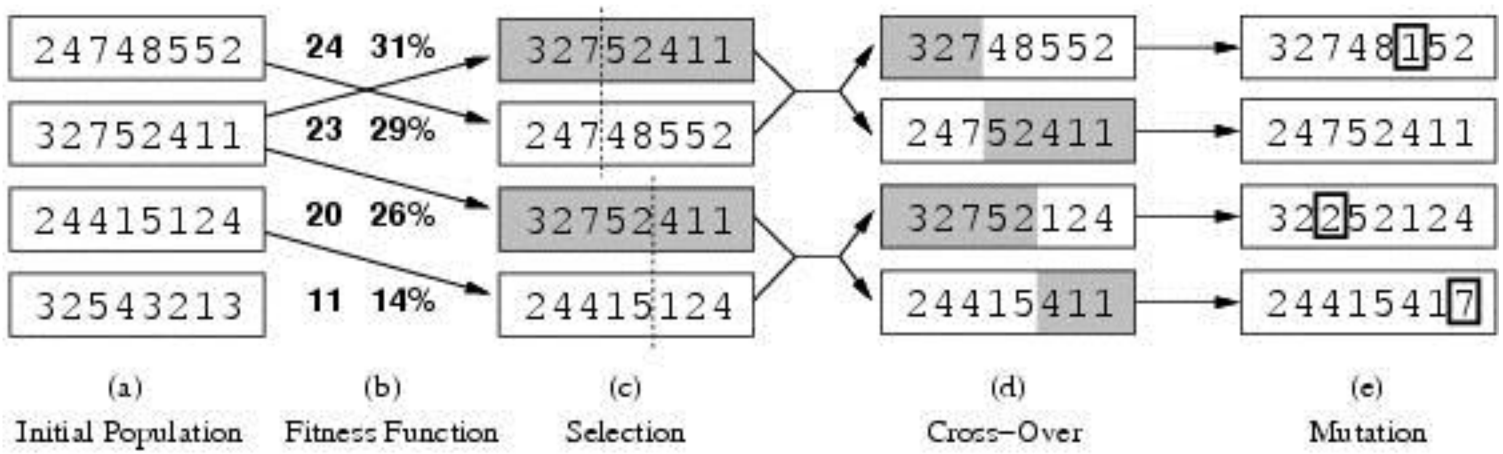


The 8-queens states involved in this reproduction steps are illustrated above.

Note: when two parents are quite different the crossover operation can produce a state that is a long way from either parent state.

It is often the case that the population is quite diverse early on in the process, so crossover (like simulated annealing) frequently takes large steps in the state space early in the search process and smaller steps later on when the individuals are quite similar.

Genetic algorithms: Production of next generation



(e) Each location is subject to a random mutation with a small independent probability.

- Here one digit was mutated in the first, third and fourth offspring.
- In the 8-queens problem this is equivalent to choosing a queen at random and moving it to a random square in its column.

Steps in the GA development

1. Specify the problem, define constraints and optimum criteria;
2. Represent the problem domain as a chromosome;
3. Define a fitness function to evaluate the chromosome performance;
4. Construct the genetic operators;
5. Run the GA and tune its parameters.

Summary

Genetic algorithms are variant of stochastic beam search algorithms

Encoding an individual

Evaluating fitness

Creating a new individual using selection, crossover and mutation

Questions?